Jointly Learning Time Series and Text Modalities for Better Human Explainability

Xuan-Hong Dang, Syed Yousaf Shah, Petros Zerfos
IBM Watson Research
Yorktown Heights, New York 10598
xuan-hong.dang@ibm.com, {syshah,pzerfos}@us.ibm.com

Abstract

Multimodal analysis that uses numerical time series and textual corpora as input data sources is becoming a promising approach, especially in the financial industry. However, the main focus of such analysis has been on achieving high prediction accuracy and less efforts been spent on using textual information as a means for explaining the behavior of the time series in human-understandable terms. In this work, we propose a novel, multi-modal, neural network model called MSIN that jointly learns both numerical time series and categorical text articles in order to discover the association. Through multiple steps of data interrelation between the two data modalities, MSIN learns to focus on a small subset of text articles that best associates with the performance in the time series. This succinct set is timely discovered and outputted as recommended documents for the given time series. We empirically evaluate our model on daily news articles collected from Thomson Reuters, using stock prices from Apple and Google as the time series. The experimental results demonstrate that MSIN achieves up to 84.9% and 87.2% in recalling the ground truth articles respectively to the two examined time series, far more superior to state-of-the-art algorithms that rely on conventional attention mechanism in deep learning.

Introduction

Current multimodal analysis that combines time series with text data often focuses on extracting features from text corpus and incorporates them into a forecasting model for enhancing prediction. Little attention is paid to the aspect of using text as a means of explaination for the patterns observed in the time series (Akita and others 2016; Weng, Ahmed, and Megahed 2017). In many emerging applications, given a time series, one can ask for finding a small set of documents that can reflect or influence the time series. Taking "quantamental investing" (Wigglesworth 2018) as an example. When trading a stock, investors do not solely base their decisions on its historical prices. Rather, the decisions are made with a careful consideration of the news and events collected from the markets. With the dramatically increasing amount of available news nowadays (Schumaker et al. 2012), a natural question hence to ask is what would be the most relevant news associated with a particular stock series. Certainly, for different stock time series, the set of relevant associated news articles would be different. Hence an automated system becomes more relevant. Likewise with the cloud business, accurately associating a cloud monitoring metric (time series) with textual complaints from customers can help technicians better understand the possible issues with the cloud system, reducing efforts spent in searching for and addressing the issues.

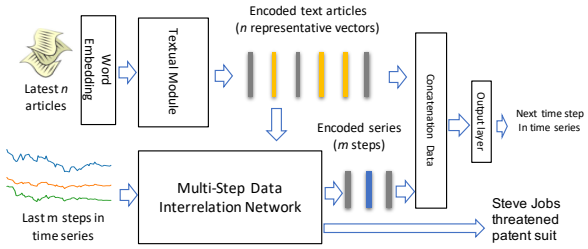

Figure 1: Our model learns to discover top relevant text articles timely associated with the current state in a given time series.

In this paper, we address the above mentioned important yet challenging problem through developing a novel, multimodal, neural network that jointly learns numerical time series and textual documents in order to discover the relation between them. The discovered text articles are returned as a means of recommended documents for the current state of the time series. As shown in Fig.1, our model consists of (1) a textual module that learns representative vectors for input text documents, (2) the MSIN (Multi-Step Interrelation Network), which takes as input both the time series and the sequence of textual representative vectors to learn the association between them, (3) a concatenation and dense layers that aggregate information from the two data modalities to output the next value in the time series.

The novelty from our proposed model stems from the introduction of Multi-Step Interrelation Network (MSIN), which allows the incorporation of semantic information learned from the textual modality to every time step modeling of the time series. The alignment of time steps or sampling frequency between time series and textual modalities is not required, allowing the sequence lengths of two modalities to be arbitrary and different. Compared to other multimodal approaches that learn two data modalities either sequentially or in parallel, MSIN leverages the mutual information impact between the two data modalities through *multi-step interrelation* of the textual representative vectors and the time

series' values. As a result, it gradually assigns large weight to "important" text segments, while rules out less relevant ones, and finally generates a textual attention that best aligns with the current state of the time series. We perform detailed empirical analysis over large-scale financial news and stock prices datasets spanning over 7 years, and show that MSIN achieves strong performance of 84.9% & 87.2% in recalling ground-truth relevant news w.r.t. two company-specific time series.

Learning text articles representation

The first network component in our model is the *textual module* that learns to represent text articles as numerical vectors so that they are comparable to the numerical time series. The order among words within each text article is important to learn their semantic dependencies. Hence, our network exploits the long-short term memory (LSTM) (Hochreiter and Schmidhuber 1997) to learn such dependencies and aggregates them into an article's representative vector.

At each time stamp t, an input data sample to the textual module is a sequence of n text documents $\{doc_1, doc_2, \ldots, doc_n\}$ (e.g., n news articles released at day t by Thomson Reuters). And its output sample is a sequence n representative vectors denoted $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$ (notation t $\mathbf{s}_j$'s is omitted to minimize clutter). Each text document j (with $1 \leq j \leq n$) in turn is a sequence of K words denoted by $doc_j = \{\mathbf{x}_1^{txt}, \mathbf{x}_2^{txt}, \ldots, \mathbf{x}_K^{txt}\}$ (txt denotes for text modality). Each $\mathbf{x}_\ell^{txt} \in \mathbb{R}^V$ is the one-hot-vector representation of the ℓ-th word, with V as the vocabulary size. We use an embedding layer to transform each $\mathbf{x}_\ell^{txt}$ into a low dimensional dense vector $\mathbf{e}_\ell \in \mathbb{R}^{d_w}$ via a linear transformation:

$$\mathbf{e}_\ell = \mathbf{W}_e * \mathbf{x}_\ell^{txt}, \quad \text{in which} \quad \mathbf{W}_e \in \mathbb{R}^{d_w \times V} \tag{1}$$

Often, d_w is much smaller than V, and $\mathbf{W}_e$ can be trained from scratch; however, using or initializing it with pre-trained vectors from GloVe (Pennington and others 2014) can produce more stable results. Also, in our examined datasets (see Section), we found that setting $d_w = 50$ is sufficient given the vocabulary size $V = 5000$.

The sequence of embedded words $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_K\}$ for an article is then fed into an LSTM that learns to produce encoded contextual vectors. The key components of an LSTM unit are the memory cell which preserves essential information of the input sequence through time, and the non-linear gating units that regulate the information flow in and out of the cell. At each step ℓ (corresponding to ℓ-th word) in the input sequence, LSTM takes in the embedding $\mathbf{e}_\ell$, its previous cell state $\mathbf{c}_{\ell-1}^{txt}$, and the previous output vector $\mathbf{h}_{\ell-1}^{txt}$, to update the memory cell $\mathbf{c}_\ell^{txt}$, and subsequently outputs the word representation $\mathbf{h}_\ell^{txt}$ for $\mathbf{e}_\ell$. From this view, we can briefly represent LSTM as a recurrent function f as follows:

$$\mathbf{h}_\ell^{txt} = f(\mathbf{e}_\ell, \mathbf{c}_{\ell-1}^{txt} \mathbf{h}_{\ell-1}^{txt}), \quad \text{for} \quad \ell = 1, \ldots, K \tag{2}$$

in which the memory cell $\mathbf{c}_\ell^{txt}$ is updated internally. Both $\mathbf{c}_\ell^{txt}, \mathbf{h}_\ell^{txt} \in \mathbb{R}^{d_h}$ with d_h is the number of hidden neurons.

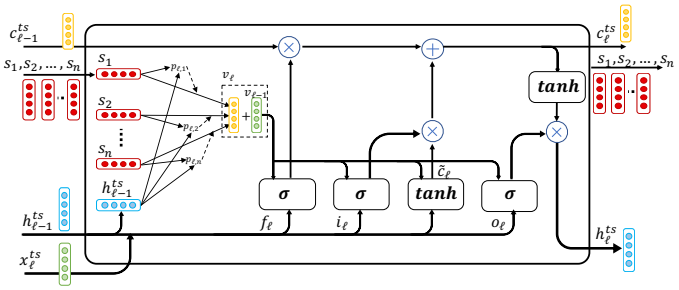

Figure 2: Memory cell design of the multi-step interrelation network (MSIN).

Our implementation of LSTM closely follows the one presented in (Zaremba and others 2014) with two extensions. First, in order to better exploit the semantic dependencies of the current word with both its preceding and following contexts, we build two LSTMs respectively taking the sequence in the forward and backward directions (denoted by head arrows in Eq.(3)). This results in a bidirectional LSTM (BiLSTM):

$$\overrightarrow{\mathbf{h}}_\ell^{txt} = \overrightarrow{f}(\mathbf{e}_\ell, \mathbf{c}_{\ell-1}^{txt}, \overrightarrow{\mathbf{h}}_{\ell-1}^{txt}), \overleftarrow{\mathbf{h}}_\ell^{txt} = \overleftarrow{f}(\mathbf{e}_\ell, \mathbf{c}_{\ell-1}^{txt}, \overrightarrow{\mathbf{h}}_{\ell-1}^{txt})$$
$$\mathbf{h}_\ell^{txt} = [\overrightarrow{\boldsymbol{h}}_\ell^{txt}, \overleftarrow{\mathbf{h}}_\ell^{txt}] \quad \text{for} \quad \ell = 1, \ldots, K \tag{3}$$

The concatenated vector $\mathbf{h}_\ell^{txt}$ leverages the context surrounding the ℓ-th word and hence better characterizes its semantics as compared to the embedding vector $\mathbf{e}_\ell$ which ignores the local context in the input sequence. Second, we extend the model by exploiting the weighted mean pooling from all vectors $\{\mathbf{h}_1^{txt}, \mathbf{h}_2^{txt}, \ldots, \mathbf{h}_K^{txt}\}$ to form the overall representation $\mathbf{s}_j$ of the entire j-th text article:

$$\mathbf{s}_j = \frac{1}{K} \sum_\ell \beta_\ell * \mathbf{h}_\ell^{txt}$$

$$\text{where} \quad \beta_\ell = \frac{\exp(\mathbf{u}^\top \tanh(\mathbf{W}_\beta * \mathbf{h}_\ell^{txt} + \mathbf{b}_\ell))}{\sum_\ell \exp(\mathbf{u}^\top \tanh(\mathbf{W}_\beta * \mathbf{h}_\ell^{txt} + \mathbf{b}_\ell))} \tag{4}$$

in which $\mathbf{u} \in \mathbb{R}^{2d_h}, \mathbf{W}_\beta \in \mathbb{R}^{2d_h \times 2d_h}$ are respectively referred to as the parameterized context vector and matrix whose values are jointly learnt with the BiLSTM. This pooling technique resembles the successful ones presented in (Conneau and others 2017) that learn multiple views over each input sequence. Our implementation simplifies them by adopting only a single view ($\mathbf{u}$ vector) with the assumption that each new story contains only one topic relevant to the time series. Note that, similar to convolutional neural networks, a max pooling can also be used in replacement for the mean pooling in defining $\mathbf{s}_j$. We, however, attempt to keep the model simple since using max function will add another level of nonlinearity and generally requires more training data to learn a proper transformation. We apply our text module BiLSTM to every text article collected at time period t and it output is a sequence of representative vectors $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$, each corresponds to one text article at input.

Multi-step interrelation of data modalities

Our next task is to model the time series, taking instant information learnt from the textual modality to discover a succinct

set of relevant text articles well aligning with the current state of the time series. A straightforward approach of using a single alignment between the two data modalities (empirically evaluated in Section) generally does not work effectively since the text articles are highly general, contain noise, and especially we do not have step-wise synchronization between them. To tackle these challenges, we propose the novel MSIN network that upgrades the LSTM so that it can handle two input sequences of different lengths. Most importantly, MSIN leverages the mutual information between the two data modalities through multi-steps of interrelation between the representative textual vectors and the time series sequence. Doing so allows MSIN to gradually filter out irrelevant text articles while focusing on only the ones that correlate with the patterns learnt from the time series, as it advances in the series sequence. The chosen text articles are finally captured in a probability mass of attention on the textual representative vectors.

Fig.2 illustrates a memory cell of our MSIN. Unlike conventional LSTMs, inputs to MSIN at each time stamp t are two sequences: (1) values of last m steps in the time series $\{\mathbf{x}^{ts}_{t-m}, \mathbf{x}^{ts}_{t-m-1}, \ldots, \mathbf{x}^{ts}_{t-1}\}$ (ts denotes for the time series modality); (2) a sequence of n text representative vectors learnt by text module $\{\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_n\}$. Its outputs are the set of hidden state vectors (described below) and the probability mass vector p_m at the last state of the series sequence, showing the text articles relevant to the time series sequence. The number of gates in MSIN's memory cell remains fixed but we *augment* the information flows within the cell by the information learnt in the text modality. To be concrete, MSIN starts with the initialization of the initial cell state $\mathbf{c}^{ts}_0$ and hidden state $\mathbf{h}^{ts}_0$ by using two separate single layer neural networks applied on the average state of the text sequence:

$$\mathbf{c}^{ts}_0 = \tanh(\mathbf{U}_{c_0} * \bar{\mathbf{s}} + \mathbf{b}_{c_0}) \quad (5)$$

$$\mathbf{h}^{ts}_0 = \tanh(\mathbf{U}_{h_0} * \bar{\mathbf{s}} + \mathbf{b}_{h_0}) \quad (6)$$

where $\bar{\mathbf{s}} = 1/n \sum_j \mathbf{s}_j$; and $\mathbf{U}_{c_0}, \mathbf{U}_{h_0} \in \mathbb{R}^{2d_h \times d_s}$, $\mathbf{b}_{c_0}, \mathbf{b}_{h_0} \in \mathbb{R}^{d_s}$, with d_s as the number of neural units. These are parameters jointly trained with our entire model.

MSIN incorporates information learnt in the text articles to every step it performs reasoning on the time series in a selective manner. Specifically, at each timestep ℓ in the time series sequence, MSIN searches through the text representative vectors to assign higher probability mass to those that better align with the signal it models so far in the time series sequence, captured in the hidden state $\mathbf{h}^{ts}_{\ell-1}$. In particular, the attention mass associated with each text representative vector $\mathbf{s}_j$ is computed at ℓ-th timestep as follows:

$$a_{\ell,j} = \tanh(\mathbf{W}_a * \mathbf{h}^{ts}_{\ell-1} + \mathbf{U}_a * \mathbf{s}_j + \mathbf{b}_a) \quad (7)$$

$$p_\ell = softmax(\mathbf{v}_a^T [a_{\ell,1}, a_{\ell,2}, \ldots, a_{\ell,n}]) \quad (8)$$

where $\mathbf{W}_a \in \mathbb{R}^{d_s \times d_s}$, $\mathbf{U}_a \in \mathbb{R}^{2d_h \times d_s}$, $\mathbf{b}_a \in \mathbb{R}^{d_s}$ and $\mathbf{v}_a \in \mathbb{R}^n$. The parametric vector $\mathbf{v}_a$ is learnt to transform each alignment vector $a_{\ell,j}$ to a scalar and hence, by passing through the softmax function, p_ℓ is the probability mass distribution over the text representative sequence. We would like the information from these vectors, scaled proportionally by their probability mass, to immediately impact the learning process over the time series. This is made possible through generating a context vector $\mathbf{v}_\ell$:

$$\mathbf{v}_\ell = \frac{1}{2}(\sum_j p_{\ell,j} * \mathbf{s}_j + \mathbf{v}_{\ell-1}) \quad (9)$$

in which $\mathbf{v}_0$ is initialized as a zero vector. As designed, MSIN constructs the latest context vector as the average information between the current representation of relevant text article (1st term on the right hand side of Eq.(9)) and the previous context vector $\mathbf{v}_{\ell-1}$. By induction, influence of context vectors in the early time steps is fading out as MSIN advances in the time series sequence. MSIN uses this aggregated vector to regulate the information flows to all its input, forget, output gates, and the candidate cell state:

$$\mathbf{i}_\ell = \sigma(\mathbf{U}_{ix} * \mathbf{x}^{ts}_\ell + \mathbf{U}_{ih} * \mathbf{h}^{ts}_{\ell-1} + \mathbf{U}_{iv} * \mathbf{v}_\ell + \mathbf{b}_i)$$

$$\mathbf{f}_\ell = \sigma(\mathbf{U}_{fx} * \mathbf{x}^{ts}_\ell + \mathbf{U}_{fh} * \mathbf{h}^{ts}_{\ell-1} + \mathbf{U}_{fv} * \mathbf{v}_\ell + \mathbf{b}_f)$$

$$\mathbf{o}_\ell = \sigma(\mathbf{U}_{ox} * \mathbf{x}^{ts}_\ell + \mathbf{U}_{oh} * \mathbf{h}^{ts}_{\ell-1} + \mathbf{U}_{ov} * \mathbf{v}_\ell + \mathbf{b}_o)$$

$$\tilde{\mathbf{c}}^{ts}_\ell = \tanh(\mathbf{U}_{cx} * \mathbf{x}^{ts}_\ell + \mathbf{U}_{ch} * \mathbf{h}^{ts}_{\ell-1} + \mathbf{U}_{cv} * \mathbf{v}_\ell + \mathbf{b}_c)$$

where $\mathbf{U}_{\bullet x} \in \mathbb{R}^{D \times d_s}$, $\mathbf{U}_{\bullet h} \in \mathbb{R}^{2d_h \times d_s}$, $\mathbf{U}_{\bullet v} \in \mathbb{R}^{n \times d_s}$ and $\mathbf{b}_\bullet \in \mathbb{R}^{d_s}$. Let $\odot$ denote the Hadamard product, the current cell and hidden states are then updated in the following order:

$$\mathbf{c}^{ts}_\ell = \mathbf{f}_\ell \odot \mathbf{c}^{ts}_{\ell-1} + \mathbf{i}_\ell \odot \tilde{\mathbf{c}}^{ts}, \quad \mathbf{h}^{ts}_\ell = \mathbf{o}^{ts}_\ell \odot \tanh(\mathbf{c}^{ts}_\ell)$$

By tightly integrating the information learnt in the textual modality to every step in modeling the time series, our network distributes burden of work in discovering relevant text articles throughout the course of the series sequence. The selected relevant articles is also immediately exploited to better learn patterns in the time series.

Output Layer: Given representative vectors learnt from the textual domain and the hidden state vectors of the time series, we use a concatenation layer to aggregate them and pass it through an output dense layer. The entire model is trained with the output as the next value in the time series. As ablation studies, we consider two variants to our proposed model. Empirically evaluating these models also highlights our model's strength in discovering text articles relevant to the time series. First, in order to see the impact of multistep of interrelation between data modalities, we exclude that process from our model, use a conventional LSTM to model the time series and subsequently align its last state with the representative textual vectors to find the information clues. This simplified model has fewer parameters yet the interaction between two data modalities is limited to only the last state of the time series sequence. We name it LSTMw/o (LSTM without interaction).

Experiment

Datasets: Our analyzed dataset consists of news headlines (text articles) daily collected from Thomson Reuters between 2006-2013 (Ding and others 2014), and the daily stock prices time series collected for the same time period from Yahoo! Finance. For the results reported below, we set $m = 5$ last days for the time series sequence, while news headlines released in

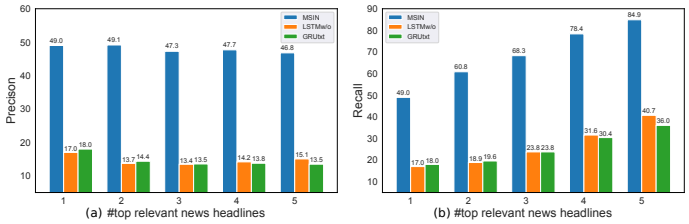

Figure 3: (a) Precision and (b) Recall computed w.r.t. "Apple" headlines annotated by Reuters.

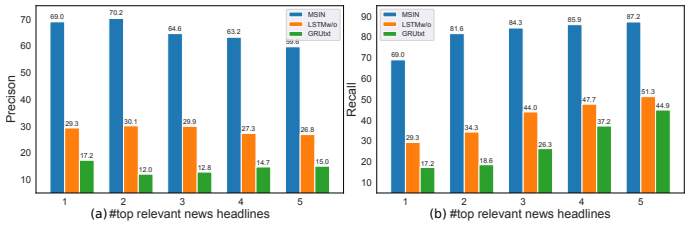

Figure 4: (a) Precision and (b) Recall computed w.r.t. "Google" headlines annotated by Reuters.

the latest 24 hours forms the text data sequence. Time stamps in both data modalities are daily basis. The datasets are split into the training, validation and test sets respectively for the year-period 2006-2011, 2012, and 2013. We use the validation set to tune the models' parameters (trained on the training set), while use the test set for independent evaluation. We evaluate the capability of our model in discovering relevant news headlines associated with two company-specific stock time series: (1) Apple (AAPL), and (2) Google (GOOG). Due to space constraints, we provide description on the model training process, detailed data description (statistics), and more comprehensive experimental results (with other time series) in the repository `https://github.com/anony-account/IAAI`. We name our model as the main network component `MSIN`, and for baseline models, we implement `LSTMw/o`, and the `GRUtxt` based on (Yang and others 2016) (results on other four baseline models are provided in the aforementioned repository).

Relevant text articles discovery

The Thomson Reuters corpus provides meta-data indicating whether a news article is about a specific company and we use such information as the ground truth relevant news, denoted by GTn. Nonetheless, it is important to emphasize that such information was not used to train our model(s). Neither the identity of time series nor pre-selection of company-specific news have been used. Rather, we let `MSIN` learn itself the association between the textual news and the stock series via jointly analyzing the two data modalities simultaneously. `MSIN` is hence completely data-driven and is straightfoward to be applied to other corpus such as Bloomberg news source, or other applications like cloud business, where similar meta-data is not available.

The GTn headlines allow us to compute the rate of discovering relevant news stories in association with a stock series through the precision and recall metrics. Higher ranking (based on attention mass) of these GTn headlines on top of each day signifies better performance of an examined model. Fig.3(a-b) and 4(a-b) plot these evaluations for the AAPL and GOOG time series respectively, when we vary the number of returned daily top relevant headlines k between 1 and 5 (shown in x-axis). For example, at $k = 5$, `MSIN` achieves up to 84.9% and 87.2% in recall while retains the precision at 46.8% and 59.6% respectively to the GTn sets of AAPL and GOOG. Other settings of k also show that `MSIN`'s performance is far better than the competitive models. The novel design of fusing two data modalities through multiple-step data interrelations allows our model to effectively discover the complex hidden correlation between the time-varying patterns in the time series and a small set of corresponding information clues in the general textual data. Its precision and recall significantly outperform those of `LSTMw/o` that utilizes only one step of alignment between the two data modalities, and of `GRUtxt` which solely explores the textual domain with conventional attention mechanism in deep learning (Yang and others 2016).

Explanation on the discovered textual news

As concrete examples for qualitative evaluation, we show in Tables 1 all news headlines from three specific examined days, along with those discovered by our model (highlighted in green and by setting their cumulated probability mass of attention $\geq 50\%$) as relevant news when it was trained with the AAPL series, and evaluated performance on the independent test dataset. As clearly seen on date 2013-01-22, `MSIN` gives high attention mass to *"steve jobs threatened patent suit to enforce no-hire policy"* though none of words mention the Apple company. Likewise, on 2013-09-06, *"china unicom, telecom to sell latest iphone shortly after u.s. launch"* received the 2nd highest probability mass (21%), in addition to the 1st one *"apple hit with u.s. injunction in e-books antitrust case"* (37%). Although ground truth news headlines (often containing company name keyword) have been used to quantitatively evaluate `MSIN`'s performance (Fig.3 and 4), we believe that these uncovered news headlines demonstrate the success and potential of `MSIN`, as our model is capable to unearth the news contents that never explicitly mention company names. We observe similar performance of `MSIN` when it was trained with the GOOG stock series, using the same textual news corpus as presented in Table 2. The results present the probability mass of attention of `MSIN` on three days selected from the test dataset. Note that date 2013-08-14 are deliberately shown in both Tables 1 and 2 to demonstrate that the set of relevant news are clearly dependent on which time series has been used to train our model along with the general text corpus. Their relevancy to each of two time series is obvious and intuitively interpretable.

Related work

Large number of single-modality studies analyzing either time series data or unstructured text documents have been proposed in the literature. Some of these studies are based on classical statistical methods (Wu and others 2016; Michell and others 2018) or neural networks (Qiu, Song, and Akagi 2016; Zhong and Enke 2017). Recent studies from the financial domain explore both time series of as-

Table 1: News headlines from 3 examined days in 2013 (test set). Relevant news headlines discovered by MSIN associated with AAPL stock series are blue-highlighted by setting: accumulated probability mass $\geq 50\%$. (Full set of discovered relevant news are uploaded on our repository due to space constraint)

GRUtxt	LSTMw/o	AsyncLSTM	News headlines
			Date: 2013-01-22
0.06	0.00	0.00	01. analysis no respite euro zone long rebalancing slog
0.12	0.12	0.15	02. japan government welcome boj ease step towards percent inflation
0.06	0.04	0.03	03. japan government panel need achieve budget surplus
0.07	0.10	0.21	04. instant view existing home sale fall december
0.14	0.12	0.09	05. german exporter fear devaluation round boj move
0.07	0.05	0.03	06. bank japan yet revive economy
0.07	0.06	0.04	07. home resale fall housing recovery still track
0.08	0.12	0.02	08. instant view google put up well than expect quarterly number
0.08	0.11	0.03	09. bank japan buy asset sp set new five year high
0.12	0.06	0.01	10. google fourth quarter result shine ad rate decline slows
0.12	0.03	0.00	11. banks commodity stock lift sp five year high
0.07	0.02	0.01	12. google fourth quarter result shine ad rate decline slows
0.07	0.14	0.40	13. steve jobs threaten patent suit no hire policy filing
			Date: 2013-08-14
0.11	0.03	0.00	01. france exit recession beat second quarter gdp forecast
0.10	0.14	0.00	02. euro zone performance suggests recovery sight european rehn
0.14	0.03	0.00	03. germany france haul euro zone recession
0.11	0.07	0.00	04. yellen see likely next fed chair despite summers chatter reuters poll
0.11	0.12	0.00	05. us modest recovery fed cut back qe next month reuters poll
0.04	0.06	0.03	06. j.c. penney share spike report sale improve august
0.10	0.06	0.00	07. wallstreet end down fed uncertainty data boost europe
0.06	0.02	0.02	08. analysis balloon google experiment web access
0.05	0.11	0.01	09. wallstreet fall uncertainty fed bond buying
0.06	0.09	0.76	10. apple face possible may trial e book damage
0.04	0.03	0.17	11. japan government spokesman no pm abe corporate tax cut
			Date: 2013-09-06
0.03	0.04	0.21	01. china unicom telecom sell late iphone shortly us launch
0.03	0.05	0.01	02. spain industrial output fall month july
0.05	0.05	0.00	03. french consumer confidence trade point improve outlook
0.08	0.02	0.00	04. uk industrial output flat july trade deficit widens sharply
0.04	0.02	0.00	05. boj kuroda room policy response tax hike hurt economy minute
0.05	0.12	0.05	06. wind down market street funding amid regulatory pressure
0.09	0.03	0.01	07. china able cope fed policy taper central bank head
0.04	0.05	0.02	08. instant view us august nonfarm payroll rise
0.05	0.04	0.04	09. analysis fed shift syria crisis trading strategy
0.03	0.04	0.01	10. factbox three thing learn us job report
0.05	0.04	0.09	11. g20 say economy recover but no end crisis yet
0.03	0.04	0.04	12. us regulator talk european energy price probe
0.04	0.04	0.01	13. wallstreet flat job data syria worry spur caution
0.04	0.03	0.00	14. job growth disappoints offer note fed
0.04	0.05	0.02	15. bond yield dollar fall us job data
0.05	0.05	0.37	16. apple hit us injunction e books antitrust case
0.06	0.12	0.01	17. wallstreet week ahead markets could turn choppy fed syria risk mount
0.04	0.04	0.01	18. china buy giant kazakh oilfield billion
0.05	0.04	0.01	19. italy wo n't block foreign takeover economy minister
0.05	0.03	0.00	20. china august export beat forecast point stabilization
0.12	0.06	0.01	21. wallstreet week ahead markets could turn choppy fed syria risk mount
0.04	0.03	0.02	22. india inc policymakers shape up ship
0.02	0.02	0.04	23. cost lack indonesia economy
0.03	0.02	0.01	24. mexico proposes new tax regime pemex

Table 2: News headlines from 3 examined days in 2013 (test set). Relevant news headlines discovered by MSIN associated with GOOG stock series are blue-highlighted by setting: accumulated probability mass $\geq 50\%$.

GRUtxt	LSTMw/o	AsyncLSTM	News headlines
			Date: 2013-01-09
0.23	0.34	0.00	01. short sellers circle stock confidence waver
0.26	0.52	0.87	02. google drop key patent claim microsoft
0.21	0.10	0.00	03. alcoa result lift share dollar up vs. yen
0.10	0.03	0.00	04. wallstreet rise alcoa report earnings
			Date: 2013-08-14
0.11	0.10	0.00	01. france exit recession beat second quarter gdp forecast
0.10	0.05	0.00	02. euro zone performance suggests recovery sight european rehn
0.11	0.07	0.00	03. germany france haul euro zone recession
0.10	0.11	0.00	04. yellen see likely next fed chair despite summers chatter reuters poll
0.12	0.08	0.00	05. us modest recovery fed cut back qe next month reuters poll
0.05	0.04	0.02	06. j.c. penney share spike report sale improve august
0.10	0.05	0.00	07. wallstreet end down fed uncertainty data boost europe
0.08	0.18	0.79	08. analysis balloon google experiment web access
0.09	0.08	0.00	09. wallstreet fall uncertainty fed bond buying
0.06	0.16	0.05	10. apple face possible may trial e book damage
0.04	0.08	0.02	11. japan government spokesman no pm abe corporate tax cut
			Date: 2013-08-29
0.08	0.09	0.01	01. india pm likely make statement economy friday
0.07	0.11	0.01	02. boj warns emerge market may see outflow
0.06	0.09	0.03	03. european rehn say lender step up assessment greece next month
0.03	0.06	0.20	04. china environment min suspends approval cnpc
0.05	0.12	0.08	05. italy still meet target scrap property tax say rehn
0.04	0.05	0.01	06. spain economic slump longer than thought but ease
0.06	0.04	0.02	07. india central bank consider gold trade minister
0.06	0.02	0.02	08. rupee fall front slow indian economy
0.05	0.02	0.02	09. india rupee bounce record low pm may address economy
0.04	0.05	0.01	10. exclusive india might buy gold ease rupee crisis
0.06	0.05	0.02	11. spain recession longer than thought but close end
0.05	0.05	0.01	12. india finance minister asks bank ensure credit flow industry
0.06	0.08	0.01	13. easing stimulus weigh oil next year reuters poll
0.03	0.05	0.01	14. exclusive india might buy gold ease rupee crisis
0.04	0.02	0.02	15. india rupee bounce record low government seek solution
0.03	0.02	0.37	16. china google power global drive
0.09	0.03	0.00	17. gdp growth beat forecast may boost case fed move
0.08	0.01	0.00	18. wallstreet rise economy but syria concern limit gain
0.06	0.01	0.03	19. oil dip syria action uncertain dollar rise data
0.02	0.00	0.01	20. boe carney say uncertainty rbs future end

set prices and the news articles (Schumaker et al. 2012; Weng, Ahmed, and Megahed 2017; Akita and others 2016) which are related to our work. Typically, these studies attempt to transform text from financial news into various numerical forms including news sentiment, subjective polarity, n-grams and combine them with stock data. These handcrafted features require extensive pre-processing and are also extracted independently from the time series data. A more recent model (Akita and others 2016) relies on RNNs that enable it to model stock series in their natural form and later merge them with the vector representation of textual news prior to making the market prediction. The goal of all these studies remains to improve prediction accuracy but not for the time series explanation purpose. They hence lacks the capability of providing relevant interpretation for time series based on the textual information.

Our work is also related to multi-modal deep learning studies (Baltrusaitis and others 2019) which generally can be classified into three categories: early, late, and hybrid (in-between), depending on how and at which level data from multiple modalities are fused together. In early fusion (Valada and others 2016; Zadeh et al. 2016), multi-modal data sources are concatenated into a single feature vector prior to being used as inputs for a learning model, while in late fusion, data is aggregated from the outputs of multiple models, each trained on a separate modality and fused later based on aggregation rules such as, averaged-fusion (Nojavanasghari and others 2016), tensor products (Zadeh et al. 2017), or a meta-

model like gated memory (Zadeh et al. 2018). The hybrid (in-between) fusion is the trade-off paradigm, which allows the data to be aggregated at multiple scales, yet often requiring synchronization among data modalities, such as in the synchronized gesture recognition (Neverova and others 2015; Rajagopalan and others 2016). Our model is related to the third category; yet, we deal with asynchronous multimodals of numerical time series and unstructured text and relax the constraints on the time-step synchronization between modalities. More significantly, we perform data fusion through multiple steps and at the low-level features which strengthens our model in learning associated patterns across data modalities.

Conclusion

Jointly learning both numerical time series and unstructured textual data is an important research endeavor to enhance our understanding of time series performance. In this work, we presented a novel neural model that is capable of discovering the top relevant textual information associated with a given time series. In dealing with the complexity of relationship between two data modalities, along with their difference in data sampling rates and lengths, we develop `MSIN` that allows the direct incorporation of information learnt in the textual modality to every time step modeling on the behavior of time series, considerably leveraging their mutual influence through time. Through this multi-step data interrelation, `MSIN` can learn to focus on a small subset of textual data that best aligns with the time series. We demonstrate the performance of our model in the financial domain using time series of two stock prices, which are trained along with a corpus of news headlines collected from Thompson Reuters. Our `MSIN` model discovers relevant news stories to the stocks that do not even explicitly mention the company name, but include highly relevant events that may influence or reflect their performance in the market.

References

[Akita and others 2016] Akita, R., et al. 2016. Deep learning for stock prediction using numerical and textual information. In *IEEE/ACIS*.

[Baltrusaitis and others 2019] Baltrusaitis, T., et al. 2019. Multimodal machine learning: A survey and taxonomy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 41(2).

[Conneau and others 2017] Conneau, A., et al. 2017. Supervised learning of universal sentence representations from natural language inference data. *arXiv preprint arXiv:1705.02364*.

[Ding and others 2014] Ding, X., et al. 2014. Using structured events to predict stock price movement: An empirical investigation. In *EMNLP*.

[Hochreiter and Schmidhuber 1997] Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation*.

[Kim 2014] Kim, Y. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*.

[Michell and others 2018] Michell, K., et al. 2018. A stock market risk forecasting model through integration of switching regime, anfis and garch techniques. *Applied Soft Computing* 67:106–116.

[Neverova and others 2015] Neverova, N., et al. 2015. Moddrop: adaptive multi-modal gesture recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 38(8).

[Nojavanasghari and others 2016] Nojavanasghari, B., et al. 2016. Deep multimodal fusion for persuasiveness prediction. In *ACM International Conference on Multimodal Interaction*.

[Pennington and others 2014] Pennington, J., et al. 2014. Glove: Global vectors for word representation. In *EMNLP*.

[Qiu, Song, and Akagi 2016] Qiu, M.; Song, Y.; and Akagi, F. 2016. Application of artificial neural network for the prediction of stock market returns. *Chaos, Solitons & Fractals* 85.

[Rajagopalan and others 2016] Rajagopalan, S. S., et al. 2016. Extending long short-term memory for multi-view structured learning. In *ECCV*.

[Schumaker et al. 2012] Schumaker, R. P.; Zhang, Y.; Huang, C.-N.; and Chen, H. 2012. Evaluating sentiment in financial news articles. *Decision Support Systems* 53(3):458–464.

[Valada and others 2016] Valada, A., et al. 2016. Deep multispectral semantic scene understanding of forested environments using multimodal fusion. In *International Symposium on Experimental Robotics*. Springer.

[Weng, Ahmed, and Megahed 2017] Weng, B.; Ahmed, M. A.; and Megahed, F. M. 2017. Stock market one-day ahead movement prediction using disparate data sources. *Expert Systems with Applications* 79:153–163.

[Wigglesworth 2018] Wigglesworth, R. 2018. The rise of quantamental investing: Where man and machine meet. *Financial Times*.

[Wu and others 2016] Wu, L., et al. 2016. Grey double exponential smoothing model and its application on pig price forecasting in china. *Applied Soft Computing* 39.

[Yang and others 2016] Yang, Z., et al. 2016. Hierarchical attention networks for document classification. In *Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*.

[Zadeh et al. 2016] Zadeh, A.; Zellers, R.; Pincus, E.; and Morency, L.-P. 2016. Multimodal sentiment intensity analysis in videos: Facial gestures and verbal messages. *IEEE Intelligent Systems* 31(6):82–88.

[Zadeh et al. 2017] Zadeh, A.; Chen, M.; Poria, S.; Cambria, E.; and Morency, L.-P. 2017. Tensor fusion network for multimodal sentiment analysis. *arXiv preprint arXiv:1707.07250*.

[Zadeh et al. 2018] Zadeh, A.; Liang, P. P.; Mazumder, N.; Poria, S.; Cambria, E.; and Morency, L.-P. 2018. Memory fusion network for multi-view sequential learning. In *AAAI*.

[Zaremba and others 2014] Zaremba, W., et al. 2014. Recurrent neural network regularization. *arXiv preprint arXiv:1409.2329*.

[Zhang and Wallace 2015] Zhang, Y., and Wallace, B. 2015. A sensitivity analysis of (and practitioners' guide to) convo-

lutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

[Zhong and Enke 2017] Zhong, X., and Enke, D. 2017. Forecasting daily stock market return using dimensionality reduction. *Expert Systems with Applications* 67:126–139.

Supplementary

Supp. Thomson Reuters Data

Fig. 5 plots the distribution of numbers of news headlines released per day by Reuters. As a skewed distribution (mode being around 12 headlines per day), we limit all models to explore up to 25 news articles per day. This setting retains more than 95% of the total number of headlines in the original textual corpus. For a small portion of days having exceptionally large numbers of daily news articles (those located to the far-right of the distribution shown in Fig. 5), we keep the last 25 news headlines on each of these days. We use NLTK toolkit (available at `https://www.nltk.org/`) for tex preprocessing, keep the vocabulary size at 5000 unique words, and remove ones that are not found in the 400k words of the GloVe.

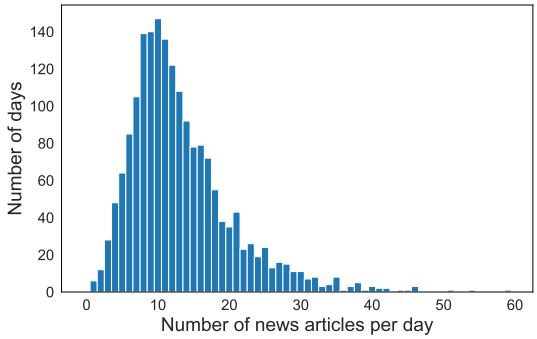

Figure 5: Distribution of numbers of news headlines released per day from the Thomson Reuters news corpus.

Supp. Statistics on Ground Truth News Headlines

Table 3: Statistics on "ground truth" news (GTn) headlines on test dataset associated with company-specific time series. %GTn shows percentage of GTn over all news headlines. %GTd shows percentage of days having at least one GTn. GTn/GTd shows average percentage of GTn on GTd. Max#GTn shows the maximum number of GTn headlines in a single day.

Dataset	%GTn	%GTd	GTn/GTd	Max#GTn
AAPL	8%	45%	15%	9
GOOG	3.5%	26%	12%	5

Supp. Parameters Setting

We set the parameter ranges for our models and its counterparts as follows: The number of layers' neural units $d_s, d_h \in \{16, 32, 64, 128\}$, regularization $L_1, L_2 \in \{0.1, 0.05, 0.01, 0.005, 0.001\}$, dropout rate $\in \{0.0, 0.1, 0.2, 0.4\}$, and the number of timesteps (sequent length) for stock time series $m \in \{3, 5, 7, 10\}$. We choose 50 as the word embedding dimension with the usage of the pre-trained GloVe model (Pennington and others 2014). For GRUtxt, we follows the model presented in (Yang and others 2016) using two Bi-LSTMs along with a self-attention layer applied at the sentence level. With CNNtxt, we implement an architecture with three kernel filters of sizes of $\{2, 3, 4\}$, using max pooling as recommended in (Kim 2014; Zhang and Wallace 2015). We use two layers of LSTMs using as encoder-decoder for the GRUts, while SVM is used with the linear kernel and C is tuned from the log range Our model and the baselines were implemented and trained on a machine with two Tesla K80 GPUs, running Tensorflow 1.3.0 as backend. We performed the random search using the validation set to tune hyperparameters.

The final values we used for AAPL dataset is $d_s, d_h = 64, m = 5, L_1 = 0.01, L_2 = 0.2$, GOOG is $d_s = 32, d_h = 64, m = 5, L_1 = 0.001, L_2 = 0.1$, both with dropout of 0.2 and using pre-trained GloVe embedding, For S&P500, the values are $d_s = 64, d_h = 96, m = 10, L_1 = 0.001, L_2 = 0.005$ with dropout of 0.1 and initializing with GloVe for word embedding.

Supp. Evaluation Metrics

In empirical evaluation of our models and its counterparts, we have used the precision and recall measurements. While their calculation with respect to predicting class labels remains as usual, their computation with respect to the ground truth news headlines (GTn) is slightly changed, being adaptive to the setting k as the number of top relevant headlines to be returned, as reported in Fig. 3(a-b), Fig.4(a-b). This is because some days have as few as only one GTn, other days may have as many as 9 (in AAPL) or 5 (in GOOG) as shown in Appendix , while the cardinality k is fixed in computing precision and recall. Specifically, we compute the Pre@k and Rec@k (reported in Fig. 3(a-b), Fig.4(a-b)) averaged from all days having at least one ground truth headline as follows:

$$\text{Pre@}k = \frac{1}{\text{GTd}} \sum_{i=1}^{\text{GTd}} \frac{TP_i[:k]}{TP_i[:k] + FP_i[:k]} \tag{10}$$

$$\text{Rec@}k = \frac{1}{\text{GTd}} \sum_{i=1}^{\text{GTd}} \frac{TP_i[:k]}{\min(k, |\text{GTn}_i[:k]|)} \tag{11}$$

in which $TP_i[:k], FP_i[:k]$ are respectively the numbers of true positive and false positive headlines at day i, and $\text{GTn}_i[:k]$ denotes the number of ground truth news headlines up to k cardinality on day i. The denominator in Rec@k ensures that the number of false negative headlines will not be beyond either the cardinality k or the actual number of ground truth headlines on day i.

Supp. Experiment on S&P500 Time Series

We show in Table 4 the headline news in three days selected from the test dataset, along with the probability mass of attention (on valid news headlines) of our `MSIN` model, its variant

Table 4: News headlines from three days selected from the test set, along with the attention mass (annotated in the numeric columns) by: GRUtxt, LSTMw/o and MSIN. Colored headlines are relevant news outputted by our model based on setting the accumulated probability mass $\geq 50\%$.

GRUtxt	LSTMw/o	MSIN	News headlines
			Date: 2013-02-03
0.00	0.53	0.00	01. china service slow uptick highlight recovery
0.08	0.10	0.00	02. japan finance minister weak yen result not goal anti deflation policy
0.02	0.01	0.18	03. france track meet percent growth target minister
0.05	0.16	0.00	04. watch central banker say not
0.12	0.01	0.08	05. us small business borrowing rise december but barely
0.28	0.08	0.07	06. gauge us business investment plan edge low
0.15	0.07	0.39	07. global share euro fall sharply renew euro zone fear
0.30	0.03	0.21	08. sp post bad day since november hill share sink
			Date: 2013-04-26
0.00	0.00	0.00	01. japan growth strategy fiscal plan g8
0.00	0.01	0.00	02. boj credibility test division emerge over inflation target
0.28	0.04	0.01	03. consumer sentiment wane april
0.00	0.01	0.05	04. economic growth gauge year high last week ecri
0.00	0.01	0.00	05. microsoft get upper hand first google patent trial
0.02	0.01	0.01	06. growth fall short forecast weakness ahead
0.22	0.09	0.14	07. dollar fall yen bond yield decline
0.28	0.80	0.73	08. wallstreet dip gdp but finish week high
0.19	0.04	0.05	09. wallstreet week ahead central bank data steer investor
			Date: 2013-10-18
0.00	0.02	0.00	01. china economy show sign slow september stats bureau
0.00	0.01	0.00	02. china share up comfort data but still weekly loss
0.00	0.08	0.01	03. china third quarter gdp growth fast year but outlook dim
0.02	0.26	0.07	04. google third quarter beat ad volume grows stock flirt level
0.01	0.11	0.03	05. schlumberger baker top estimate global drilling
0.26	0.02	0.01	06. us data boon computer driven trading
0.00	0.00	0.00	07. fed release delayed report industrial output october
0.00	0.00	0.00	08. data delay add us inflation bond sale
0.20	0.10	0.07	09. stock split wane may follow google 1,000
0.06	0.03	0.41	10. global stock gain dollar fall fed see stay course
0.44	0.31	0.39	11. sp break record google stock top earnings
0.01	0.03	0.00	12. google share break mobile pay

LSTMw/o and GRUtxt respectively in 3rd-1st columns. As observed, our model MSIN places more probability mass on the headlines whose contents directly report the market performance, while zeros out mass on headlines that report company-events or local markets. Its attention mass is also more condense, with clear focus on a small set of top relevant news, as compared to those of LSTMw/o and GRUtxt that both spread out on multiple headlines. The headlines highlighted in Table 4 are those outputted by our model by setting their cumulated probability mass of at least 50%.

The focus of our work is on discovering relevant text articles in association with the numerical time series. As using the global market time series for experiment, nevertheless, we would like to test the impact of selected relevant text articles discovered by our model in forecasting the overall market movement in the subsequent day. For this task, we further compare its performances against other baseline models, including CNNtxt (Kim 2014) that both exploit the textual news modality; GRUts that analyzes the time series; and LSTMpar, closely related to (Akita and others 2016), which trains in parallel two LSTM networks, each on one data modality, then fuses them together prior to the output layer. Moreover, for a conventional machine learning technique, we implemented SVM (Weng, Ahmed, and Megahed 2017) that takes in both time series (as vectors) and the uni-gram for the textual news.

Table 5: Performance of all models on forecasting market movement. Precision and recall are reported w.r.t. two prediction of up and down market respectively.

Models	Acc.	Pre.	Rec.
LSTMw/o	55.8	**63.9**/48.2	56.1/**56.4**
LSTMpar	53.6	57.8/43.5	70.1/31.3
GRUts	52.8	58.1/43.4	65.7/33.6
GRUtxt	55.3	60.1/45.1	70.9/33.3
CNNtxt	55.9	61.4/46.9	64.1/45.3
SVM	53.1	58.1/41.2	**71.7**/27.4
MSIN	**56.4**	63.3/**50.1**	56.9/56.3

Table 5 reports the forecasting accuracy, precision and recall of all models. Two values in each entry of precision and recall columns are respectively reported for the forecasting of up and down market on the next day respectively. As observed, there is not much difference in the prediction accuracy of GRUtxt and CNNtxt though they analyze the textual news with different network topologies. Without exploiting the temporal order in the time series and the semantic dependencies of textual news, SVM's performance seems less comparable to its counterpart neural networks. Out of all models, our MSIN achieves higher prediction accuracy. Its solution also converges to a more balanced classification boundary as reflected in the precision and recall measures.

Table 6: Performance of all models on forecasting market movement as varying the daily latest time up to which news headlines are collected.

Models	9:00	15:40	19:00	23:59
LSTMw/o	56.1	57.2	67.7	77.6
LSTMpar	54.5	56.1	65.9	76.1
GRUtxt	56.9	58.1	68.3	78.2
CNNtxt	56.8	**60.1**	69.1	78.4
SVM	54.3	55.8	61.2	75.9
MSIN	**57.1**	59.8	**69.6**	**79.3**

In order to further observe the impact of textual news on the models' classification accuracy, we vary the latest time, up to which the daily news headlines are collected, to 9:00 (before market open), 15:40 (before market close), 19:00, and 23:59 on the same day the market performance is predicted. These evaluations are reported in Table 6. A clear trend is seen that, the prediction accuracy is higher as the news articles are collected closer to the market closing time, which confirms the indicative information embedded in the textual news toward predicting the S&P500 performance. While most of neural models tend to perform feature engineering better than that of SVM, only our MSIN can further offer better interpretation due to its join-training of both evolving time series and the news articles. It is noted that evaluations at the time stamp of 19:00 (or 23:59) will shift a model from a predictive system to a purely explanatory one.